

## EXCEPTIONAL INTERRUPTION PROCESSING SYSTEM

Publication number: JP60225943

Publication date: 1985-11-11

Inventor: TSUTSUI SHIGECHIKA; SUGITA YUMIKO

Applicant: HITACHI LTD

Classification:

- international: G06F9/48; G06F9/46; G06F12/10; G06F9/46;  
G06F12/10; (IPC1-7): G06F9/46; G06F12/10

- European:

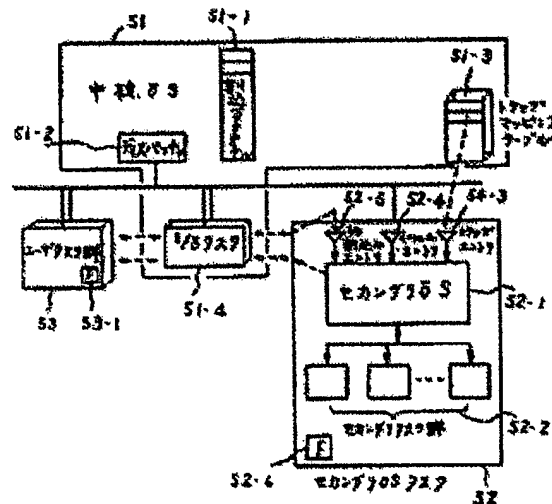
Application number: JP19840081757 19840425

Priority number(s): JP19840081757 19840425

Report a data error here

### Abstract of JP60225943

**PURPOSE:** To execute a trap processing at a high speed by constituting a tiled system so that an address of a task corresponding to a trap factor is obtained from a trap mapping table, by the contents of a trap vector table, with respect to a trap issued at the time of storage. **CONSTITUTION:** When a trap of a supervisor call SVC, etc. is generated, in case the contents of a flag register 61 are "1", it is a trap from a secondary OS task 52, therefore, first of all, the contents of a program counter PC and a status register SR are saved to a stack which is being used at present. Thereafter, the contents of a trap mapping table 63 of an address shown by a pointer register 62 are referred to directly without executing a processing for referring to an address of an interruption vector table 51-1, which is executed by a conventional technique. Subsequently, a control shifted directly to an address of a trap processing task corresponding to a trap factor is executed.



Data supplied from the esp@cenet database - Worldwide

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭60-225943

⑬ Int.Cl.<sup>4</sup>

識別記号

庁内整理番号

⑭ 公開 昭和60年(1985)11月11日

G 06 F 9/46  
12/10

G-8120-5B  
8219-5B

審査請求 未請求 発明の数 1 (全6頁)

⑮ 発明の名称 例外割込み処理方式

⑯ 特 願 昭59-81757

⑰ 出 願 昭59(1984)4月25日

⑱ 発 明 者 筒 井 茂 義 川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑲ 発 明 者 杉 田 由 美 子 川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑳ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉑ 代 理 人 弁理士 高橋 明夫 外1名

明 細 書

発明の名称 例外割込み処理方式

特許請求の範囲

1. 複数のOSを有し、割込みベクタテーブルを用いて割込み処理を行う計算機システムにおいて、該システムで実行中プログラムの属性を記憶する手段と、トラップ処理のタスクアドレスをセットしているトラップ・マッピング・テーブルを指示するトラップ・ベクタテーブルとを有し、前記記憶手段に記憶時に発行したトラップに対し、前記割込みベクタテーブルを用いる代りに、前記トラップ・ベクタテーブル内容によつて、トラップ・マッピング・テーブルからトラップ要因に対応するタスクのアドレスを得て、処理を行うことを特徴とする例外割込み処理方式。
2. 前記トラップ・ベクタテーブル内容でトラップ処理を行う場合、計算機の実行モードを変更しないことを特徴とする特許請求の範囲第1項記載の例外割込み処理方式。

発明の詳細な説明

〔発明の利用分野〕

本発明は、単一の計算機システムにおいて、特に複数のオペレーティングシステム(Operating System、以下OSという)を効率よく制御するための制御方式に関するものである。

〔発明の背景〕

従来より単一の計算機システムを多目的に利用する方法として、それぞれ目的に合致する複数のOSを作成し、それらのOSを一つの計算機システム(実計算機)の下で時分割に動作させて目的ごとのアプリケーションプログラムを実行するという仮想計算機方式が採用されている(例えば、山谷他著「仮想計算機」共立出版参照)。

第1図は、仮想計算機の構成例を示した図である。実計算機1は第1OSで動く第1仮想計算機2と、第2OSで動く第2仮想計算機3からなり、これら2つのOSが動作して、処理を行う仮想計算機である。

第2図は、上記仮想計算機システムにおける各

種プログラムの構成を示した図である。

第2図の制御プログラム(CP)21は、第1OS22および第2OS23に対する仮想環境を実現するためのプログラムで、その主な機能は、I)各仮想計算機のOSが有効に動くように仮想計算機をスケジューリングすること、II)スケジューリングに従って、仮想計算機に制御を与えること、III)外部からの割込みに対して、各仮想計算機のOSに連絡すること等であり、ユーザプログラムを直接制御する機能はない。ユーザタスク24-1...24-kからなる第1ユーザタスク群24およびユーザタスク25-1...25-nからなる第2ユーザタスク群25は、それぞれ第1OS22および第2OS23によつて制御される。

例えば、ユーザタスク24-1がOSに対する要求であるスーパーバイザ・コール(Super Visor Call、以下SVCという)割込みを送出した場合の制御例を第3図に示す。第3図に示したように、ユーザタスク24-1のSVC割込みが制御プログラム(CP)21で受理された後、第1OS

22に処理が渡され、再び制御プログラム(CP)21を経由して、ユーザタスク24-1に戻るの、第1OS22のSVC処理の他の制御プログラム(CP)21間とのオーバーヘッドが加わり、OS全体のオーバーヘッドが増加するという問題がある。

上記の問題を解決する方法に、特許願(特願昭57-62431号)「計算機システムの制御方式」昭和57年4月16日がある。

その内容を第4図に示す計算機システムの構成により説明する。

第4図において、4はプロセッサ、5は主メモリ、6は入出力装置、7はバスライン、51は中核OS、52セカンダリOSタスク、53はユーザタスク群である。

第4図に示したこの方式では、前述した制御プログラム(CP)21を設けず、中核OS51とその制御のもとに動くセカンダリOSタスク52との構成により、OS全体のオーバーヘッドを減少させている。

第4図に主メモリ5に格納されるプログラムは第5図に示す内容である。

第5図において、51-1は割込みベクタテーブル、51-2はディスパッチャ、51-3はトラップ・マップ・テーブル、51-4はI/Oタスク、52-1はセカンダリOS、52-2はセカンダリタスク群、52-3はトラップエントリ、52-4はイニシャルエントリ、52-5はI/O割込みエントリ、52-6、53-1はフラグ・レジスタである。

第5図の中核OS51は、リアルタイム・アプリケーションプログラムのユーザタスク群53を制御するもので、オーバーヘッドの小さい通常のリアルタイムOSである。その構成は、SVCトラップ、例外トラップ、外部割込みなどの割込みに対するOSへのエントリアドレスを示すテーブルで普通は計算機のメモリアドレスの若番地に設ける割込みベクタテーブル51-1と、各トラップ種別に対し、固有の処理が行えるように、各タスクごとの処理アドレスを定義するトラップ・マッ

ピング・テーブル51-3と、ディスパッチャ51-2と、I/Oタスク51-4等からなっている。中核OS51は例えばユーザタスク群53からのOS要求であるSVC割込みに対し、割込みベクタテーブル51-1で示すエントリアドレスの処理ルーチンを実行した後に、ディスパッチャ51-2を経由して、元のユーザタスクまたは、他のユーザタスクに制御を渡すという流れで動作を行う。

一方、セカンダリOSタスク52はセカンダリOS52-1、セカンダリタスク群52-2等で構成される。セカンダリOS52-1と上記OSのものに制御され、マルチプログラミングで動作するセカンダリタスク群52-2は、ユーザタスク群53と基本的に同様な位置付けで中核OS51によつて制御される。

セカンダリOS52-1への主要なエントリは、3つある。第1のエントリは、中核OS51のスケジューリングによつて、最初にディスパッチされたときの実行番地であるイニシャルエントリ

52-4である。第2のエントリは、トラップ・マッピング・テーブル51-3のアドレスで入ってくるトラップエントリ52-3である。第3のエントリは、I/Oタスク51-4からのI/O割込みエントリ52-5である。

今、セカンダリタスク群の処理実行中に、SVC、またはバスエラー・特権命令違反等によるトラップが発生した場合には、その内容で定まる中核OS51の割込みベクタテーブル51-1のアドレスに従って、中核OS51内のトラップエントリに入り、その後、トラップ・マッピング・テーブル51-3で示すSVCまたはトラップ処理用タスクのアドレスが、トラップエントリ52-3を経由して、セカンダリOS52-1に送られ、処理される。

上記のような動作の流れで処理されるので、処理時間が長くなるという問題がある。

#### 〔発明の目的〕

本発明の目的は、上記のような従来技術の問題点を解決するため、計算機システムの構成を変更

したり、あるいはOSのプログラムを複雑化することなく、OSタスクにおけるトラップの処理を高速に行うことができる例外割込み処理方式を提供することにある。

#### 〔発明の概要〕

上記目的を達成するため、本発明の例外割込み処理方式は、複数のOSを有して、割込みベクタテーブルを用いて割込み処理を行う計算機システムにおいて、該システムで実行中プログラムの属性を記憶する記憶手段と、トラップ処理のタスクアドレスをセットしているトラップ・マッピング・テーブルを指示することのできるトラップ・ベクタテーブルとを有し、前記記憶手段に記憶時に発行したトラップに対し、前記割込みベクタテーブルのかわりに、前記トラップ・ベクタテーブル内容によって、トラップ・マッピング・テーブルからトラップ要因に対応するタスクのアドレスを得て、処理を行うことに特徴がある。

#### 〔発明の実施例〕

以下、本発明の実施例を図面により説明する。

本発明における計算機システムの主メモリに格納するプログラムの構成は前述した第5図と同様である。第6図は本発明におけるレジスタの構成を示し、第7図は本発明に基づくトラップ割込み処理の動作フローチャートを示したものであり、いずれも第5図を補足説明するための図である。

第6図のフラグレジスタ61には、第5図のセカンダリOSタスク52が処理実行中には“1”、その他のタスク時には“0”がセットされ、ポイント・レジスタ62にはトラップ・マッピング・テーブル63の番地内容がセットされ、次にステータスレジスタ64 (Status Register、以下SRという) の13bit目には、スーパーバイザモード時は“1”、ユーザモード時は“0”がそれぞれセットされる。またトラップ・マッピング・テーブル63には、第5図のトラップ・マッピング・テーブル51-3と同様に、SVCなどのトラップ要因に対応する処理タスクのアドレスがセットされている。

前記フラグレジスタ61に対する“1”のセット

を、中核OS51がセカンダリOSタスク52をディスパッチングするときに行うので、計算機は実行中のタスクの属性を識別することができる。

次に、第5図および第6図の動作の流れを第7図により説明するが、必要に応じて第4図の計算機システムの構成を示す図を参照することとする。

SVCなどのトラップが発生したとき計算機は、フラグレジスタ61の内容が1または0かを判断する(71)。その内容が“0”の場合には、セカンダリOSタスク52以外からのトラップであることから、前述した通常の動作の流れ、つまりプログラム・カウンタ(以下PCという)およびSR64の内容をシステムスタックに退避し、SR64をスーパーバイザモードにセットした後、割込みベクタテーブル51-1のアドレス内容でトラップ処理を実行する動作となる(74, 75, 76)。

また、フラグレジスタ61の内容が“1”の場合には、セカンダリOSタスク52からのトラップであることから、まずPCおよびSR64の内容

を現在使用中のスタックに退避した後、従来技術で行っていた割込みベクタテーブル51-1のアドレス参照の処理をせずに、直接にポインタレジスタ62で示す番地のトラップ・マッピング・テーブル63の内容参照を行い、トラップ要因に対応するトラップ処理用タスクのアドレスに直接移行する制御となる(72, 73)。

なお、通常のトラップ処理では、トラップが発生した場合、計算機の実行モードを独立・無条件に発生以前の状態、いわゆるスーパーバイザモードにしているのに対し、本発明によるトラップ処理では、実行モードの変更を行っていない。それはフラグレジスタ61が“1”のときのセカンダリOSタスク51からのトラップ発生を中核OS51に対して報告せずに、セカンダリOSタスク51内で連続してトラップ処理を実行するためである。

また、フラグレジスタ61を独立なレジスタとして扱っているが、SR64の未使用ビットを充当するなどでも実現できる。

さらにトラップ・マッピング・テーブル63を、ポインタレジスタ62に示す番地に設けたが、固定の番地でもよく、ハードウェアレジスタ群の構成でも実現することができる。

このように、第6図に示す本発明のレジスタ構成を用いることにより、セカンダリOSタスク52からのトラップ発生を中核OS51による介入なく、高速にセカンダリOSタスク52による処理が可能となり、トラップ処理のオーバーヘッドの減少およびセカンダリOSタスク52の性能アップを計ることができる。

本発明のレジスタ構成が通常タスクすなわちフラグレジスタ61に“1”をセットしないタスクに悪影響を与えることは何もなく、また実施例で引用したところの第1OSのもとに第2OSを構成する場合に限らず、トラップをタスク内にて高速処理する場合などにも応用することができる汎用性をもつ機能である。

#### 【発明の効果】

以上説明したように、本発明の例外割込み処理

方式によれば、フラグ、ポインタ、ステータスの各レジスタを設けることにより、計算機のシステム構成を変更したり、OSで複雑なプログラムを作成したりすることなく、OSタスクにおけるトラップの処理を高速に行うことができる。

#### 図面の簡単な説明

第1図は仮想計算機の構成例を示す図、第2図は仮想計算機システムのプログラムのプログラム構成図、第3図はSVC処理動作を説明するための図、第4図は計算機システムの構成図、第5図はトラップ等の処理動作を説明するための図、第6図は本発明におけるレジスタの構成図、第7図は本発明によるトラップ処理の動作フローチャートである。

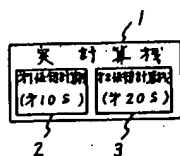
1…実計算機、2, 3…仮想計算機、4…プロセッサ、5…主メモリ、6…入出力装置、7…バスライン、21…制御プログラム(CP)、22, 23…OS、24, 25…ユーザタスク、51…中核OS、52…セカンダリOSタスク、53…ユーザタスク、61…フラグレジスタ、62…ポ

インタレジスタ、63…トラップ・マッピング・テーブル、64…ステータレジスタ。

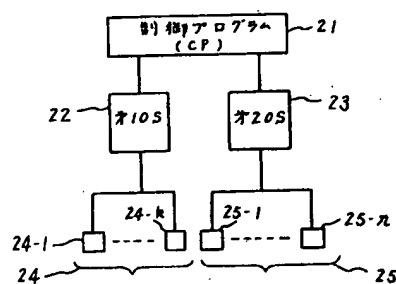
代理人 弁理士 高橋明



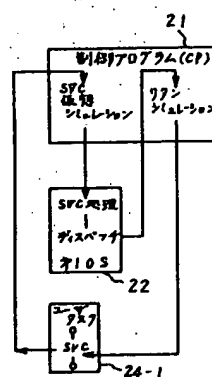
第 1 圖



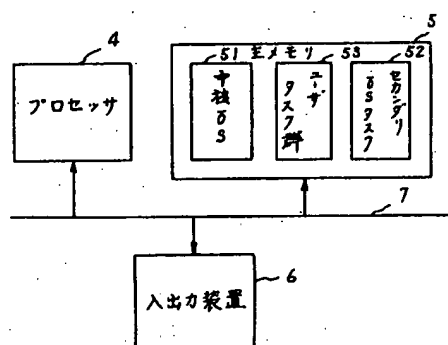
第 2 回



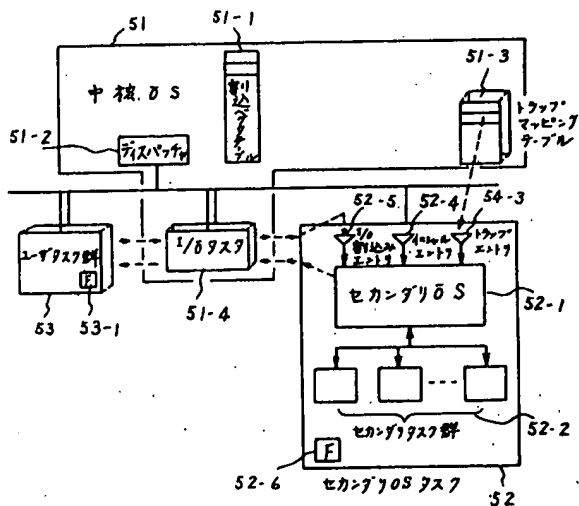
第 3 圖



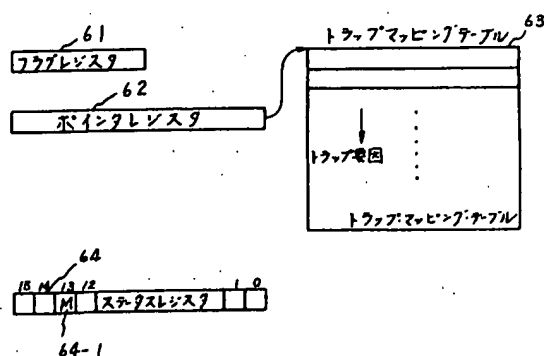
第 4 圖



第 5 圖



第 6 圖



第 7 図

